

TOSHIBA

Leading Innovation >>>

CPU affinityによる リアルタイム性向上手法

小林 良岳、山田真大

株式会社 東芝 ソフトウェア技術センター

2013/09/13

組込み機器へのOS適用課題

- 組込み機器の高性能化→汎用OSの採用
 - メモリサイズ増加
 - ディスクサイズ増加
 - multicore CPUの搭載
 - 仮想化機能の搭載
- Linux適用のメリット
 - 豊富なソフトウェア資産
 - 脆弱性発見とバグ修正のサイクル
- 適用課題
 - リアルタイム性の確保
 - 数百～数十usの割り込み遅延時間の要求

ハイブリッドOS

- LinuxとRTOSを同一HW上で動作させる手法
- リアルタイム処理はRTOSで、汎用処理はLinuxで.
- 仮想化機能を用いたり、RTOSの一タスクにLinuxを割り当てる方法

シングルOS

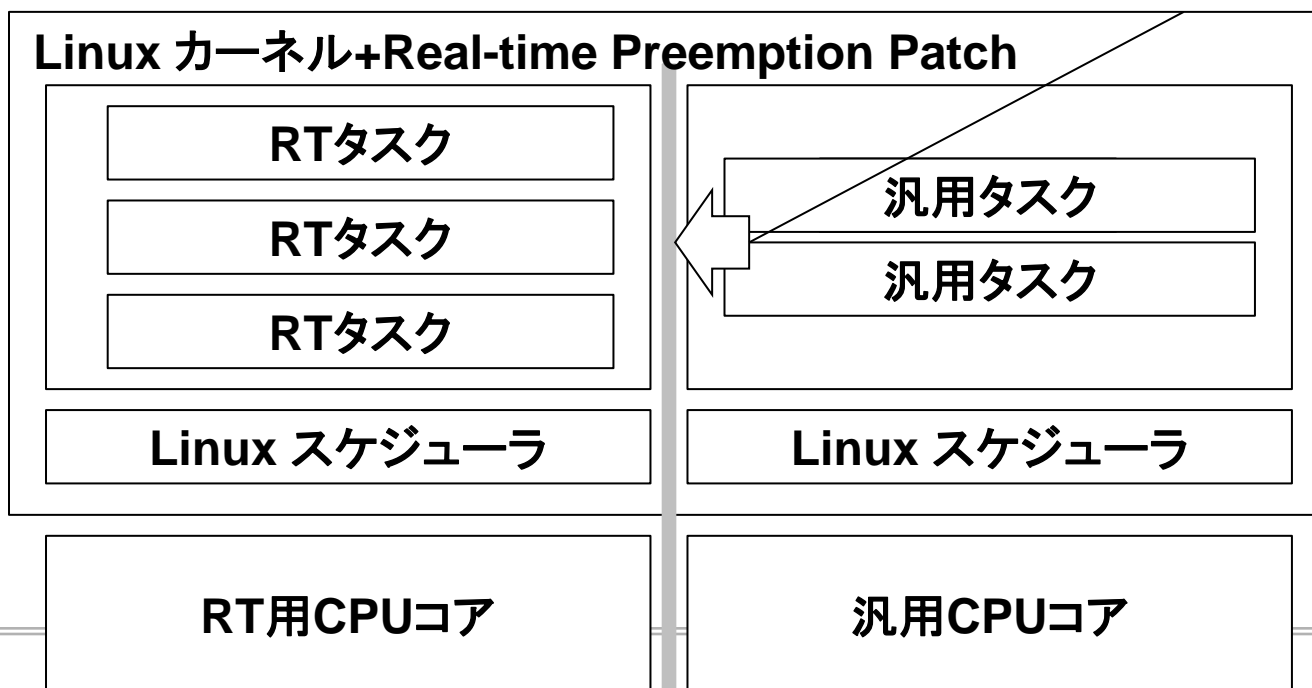
- Linux単独でリアルタイム性を向上させる／保障する手法.
- カスタマイズされたLinuxを使う
 - インタフェースも含めて大きく改変
 - インタフェースはそのまま、割り込み遅延時間を改善する方法もある.

	ハイブリッドOS	シングルOS
複雑さ	×	○
リアルタイム性	○	△

マルチコア+シングルOSのRT性向上手法

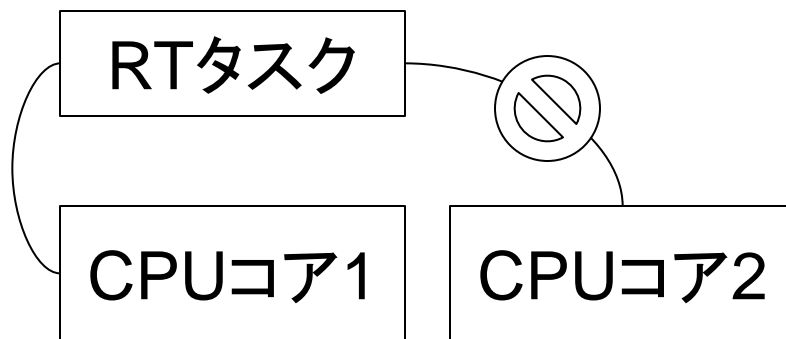
- Real-time Preemption Patch
 - 割り込み遅延時間の改善
- CPU affinity機能
 - マイグレーションが発生しない
 - マイグレーションコストや予測可能性の低下を防ぐ
 - RT性を悪化させるタスクを特定のCPUコアに限定

負荷による影響



CPU affinityの機能の効果

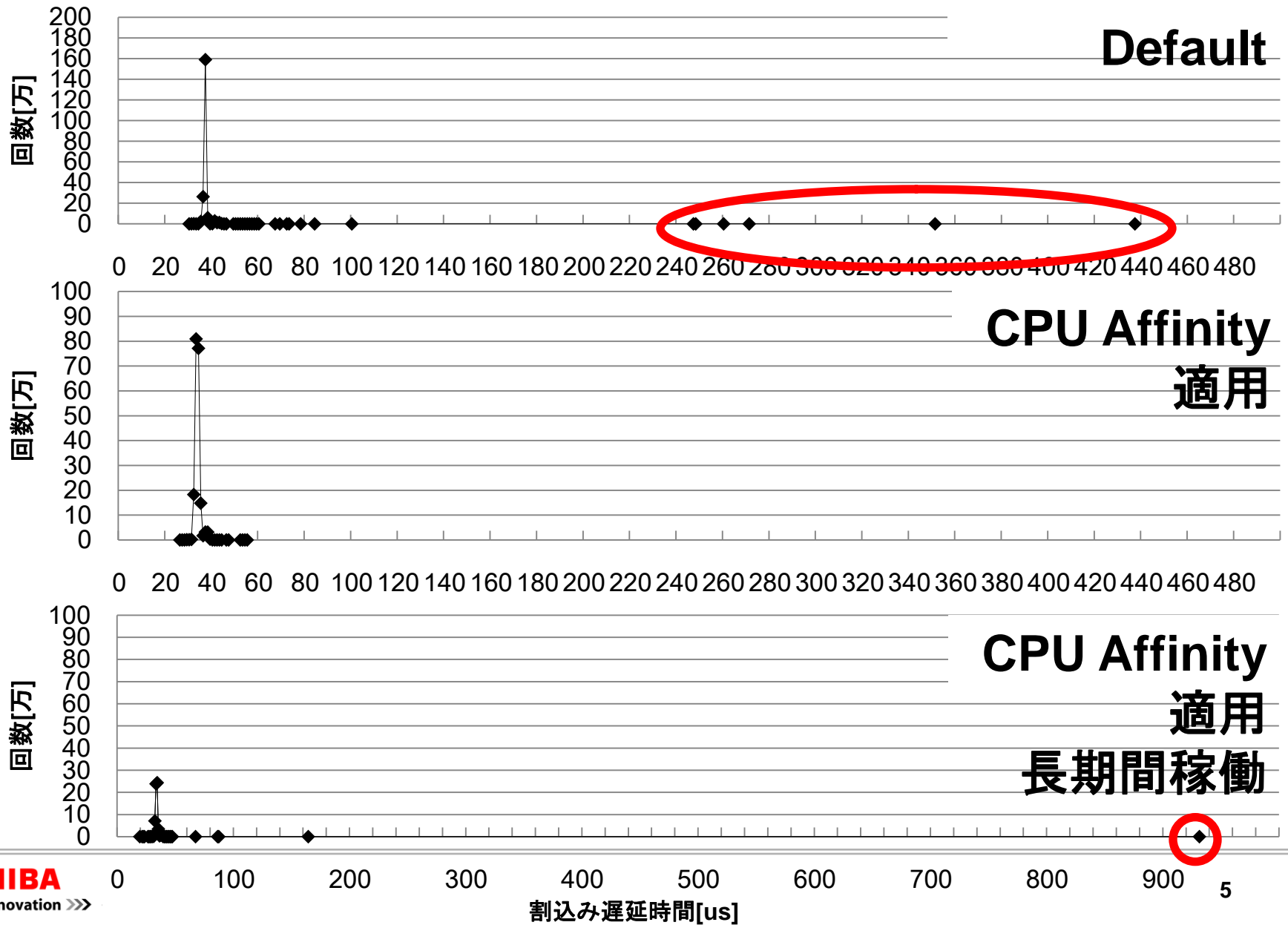
- タスク／プロセスを特定のCPUコアに限定して動作させる



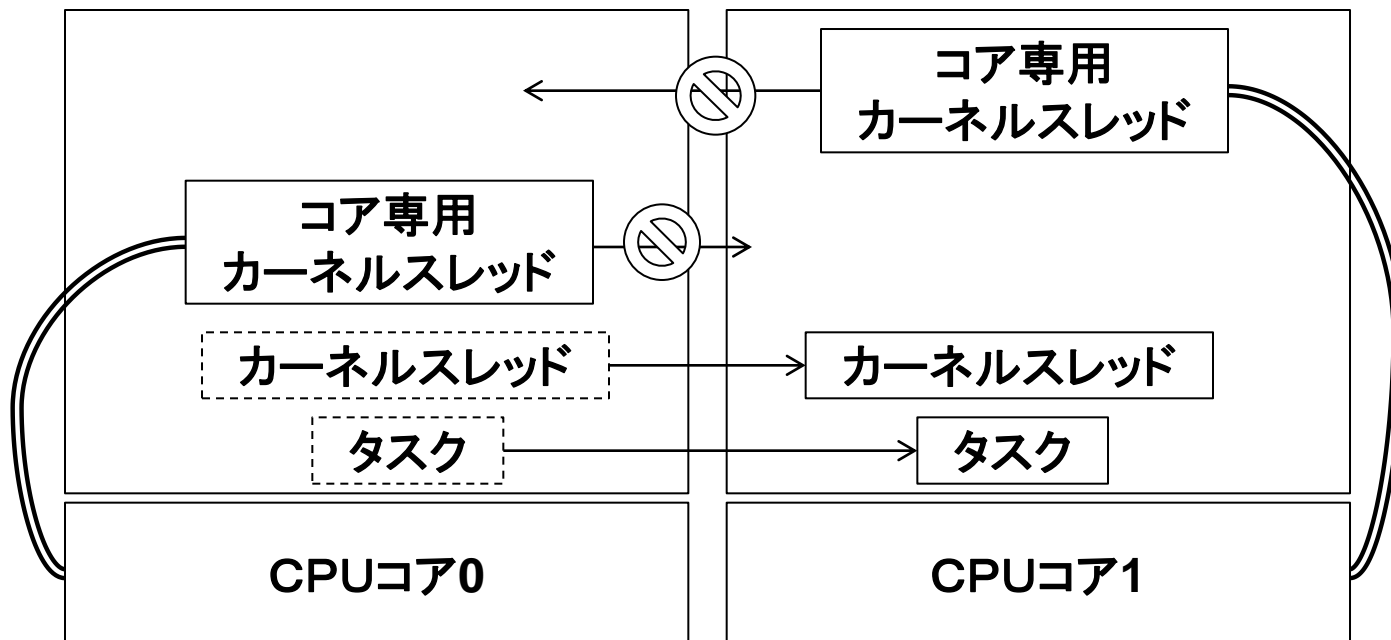
効果

- マイグレーションが発生しない
 - タスクマイグレーションはRT性と相性が悪い
 - マイグレーションコストや予測可能性の低下
- RT性を悪化させるタスクを特定のCPUコアに限定
 - RT性を必要とするCPUコアで, RT性を確保

CPU affinityの効果(割込み遅延時間評価)



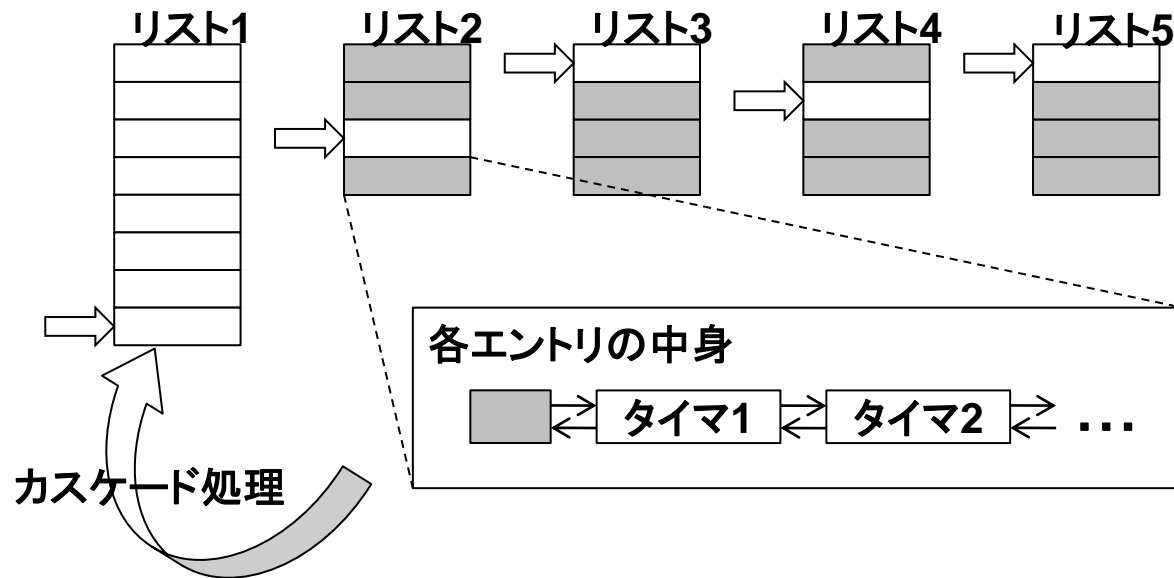
CPU affinity機能の適用範囲



■ 代表的なLinuxのコア専用カーネルスレッド

- タイマ処理, 高精度タイマ処理
- マイグレーション処理
- 遅延処理, 割込み遅延処理

タイマリストとカスケード処理



■ カスケード処理

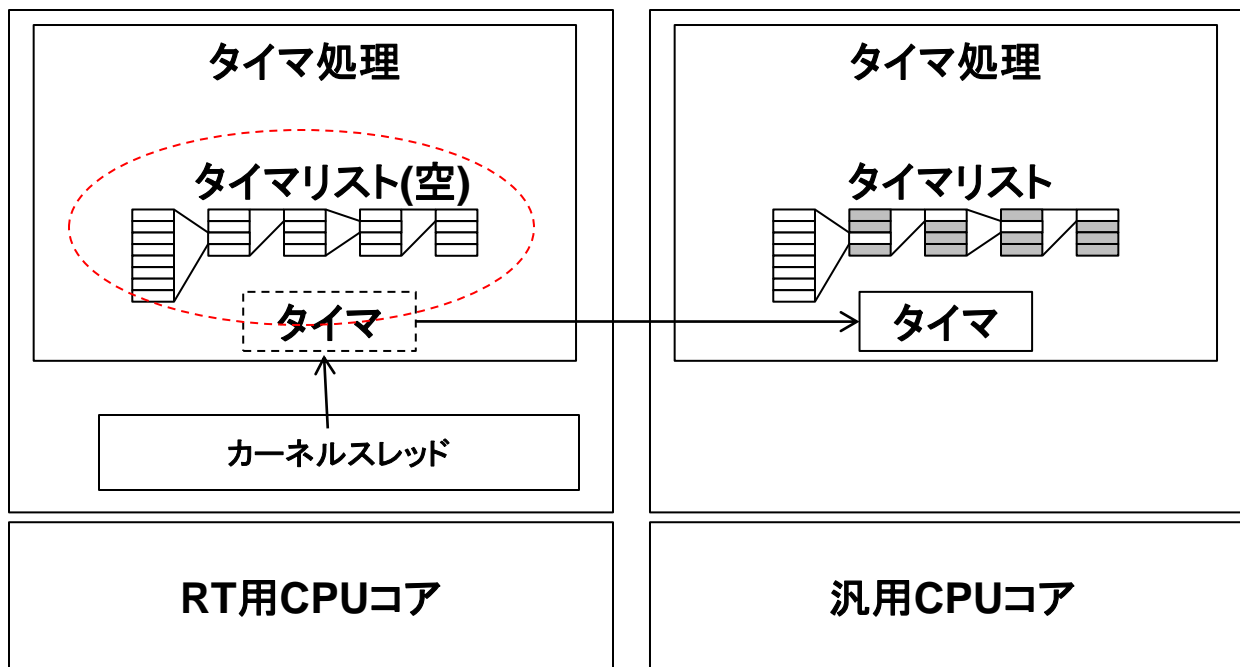
- 時刻が進み、リストnの最後のindexまで経過すると、上位のリストn+1の次のエントリに登録されているタイマをリストnに登録しなおす

■ カスケード処理の特徴→RT性の低下を招く

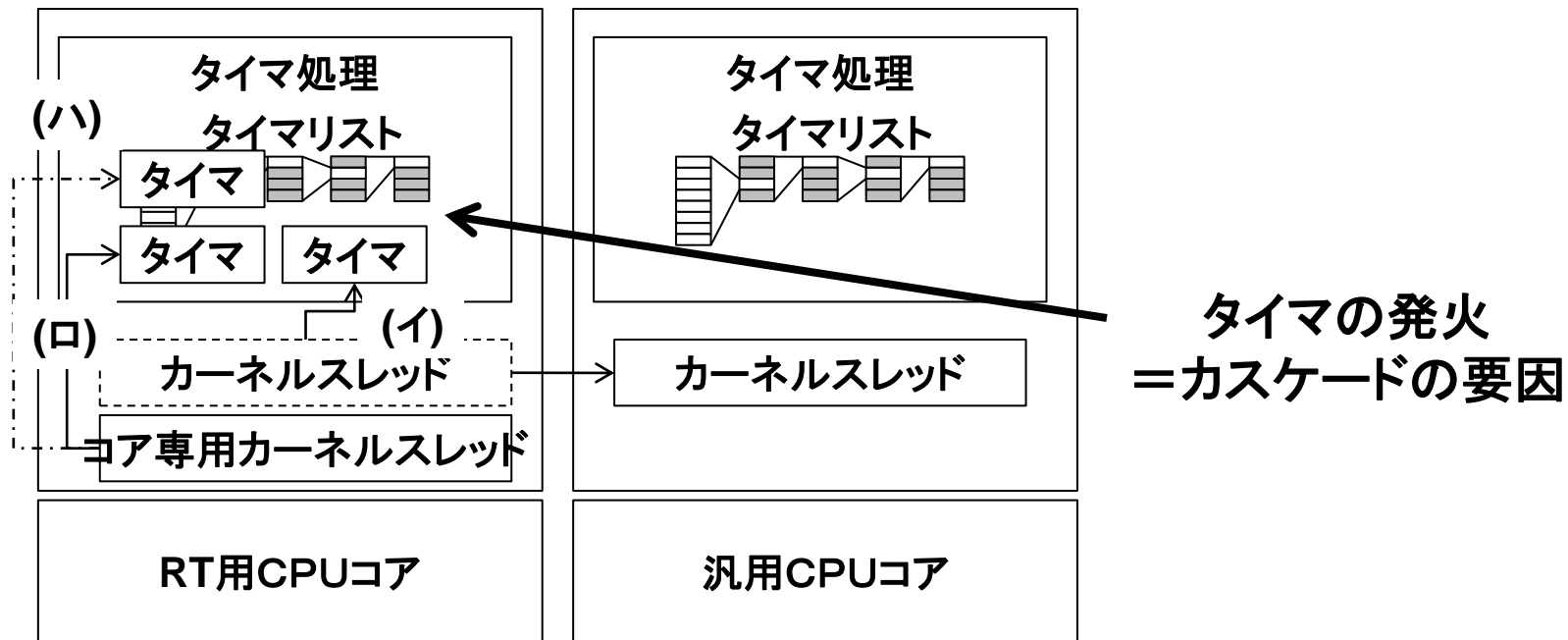
- 割込み禁止状態で実行
- タイマ数に上限がないため実行時間の予測が難しい
- ticklessカーネルの場合、一度に行うカスケード処理の実行時間が膨大

RT用CPUコア上のカスケード処理対策

- カスケード処理が発生
 - タイマが登録／発火したことが要因
- タイマリストを空の状態に維持すること
 - ⇒カスケード処理を未然に防ぐ

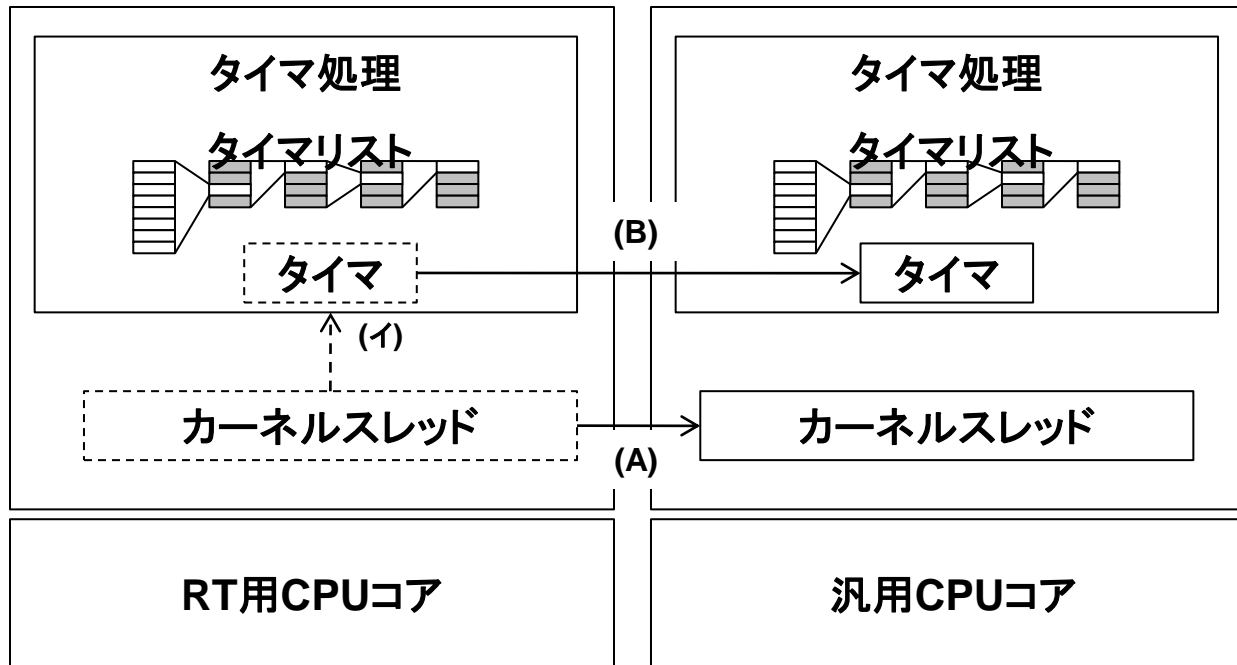


稼働状態でカスケードを引き起こす3つの課題



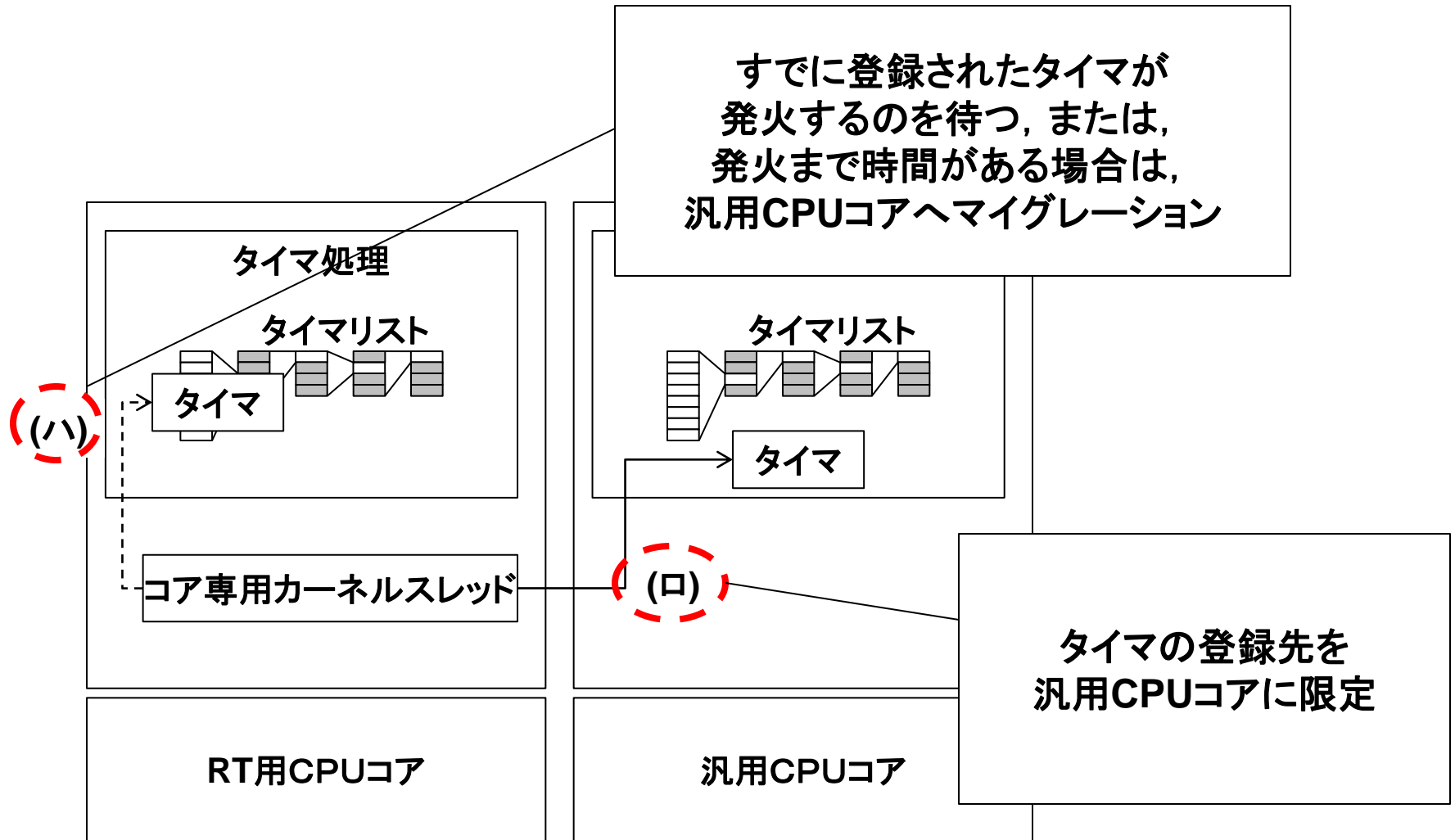
- (イ) 汎用CPUコアのタスク／カーネルスレッドが、RTタスク開始の前に登録しておいたタイマが発火する場合。
- (ロ) RT用CPUコアのコア専用カーネルスレッドが、タイマを登録し発火する場合
- (ハ) RT用CPUコアのコア専用カーネルスレッドが、RTタスク開始の前に登録したタイマが発火する場合

課題(イ)に対する提案手法

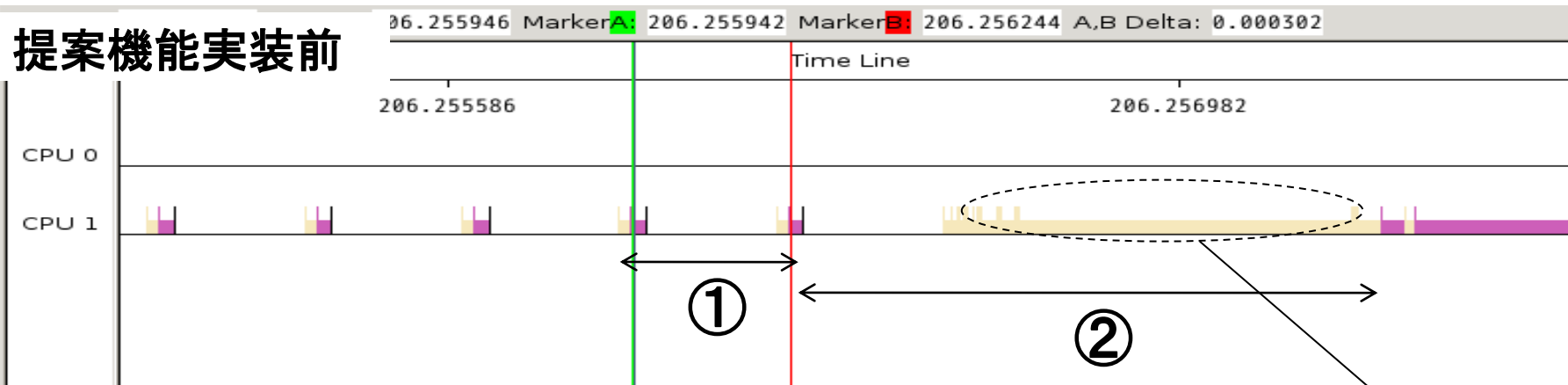


- (イ) タスク/カーネルスレッドがタイマを登録する際、履歴を残す
- (A) RT用CPUコアのカーネルスレッド／汎用タスクが汎用CPUコアへ移動.
- (B) 既に登録済み(イ)のタイマを、履歴を元に、汎用CPUコアのタイマリストへ移動(**タイママイグレーション**)

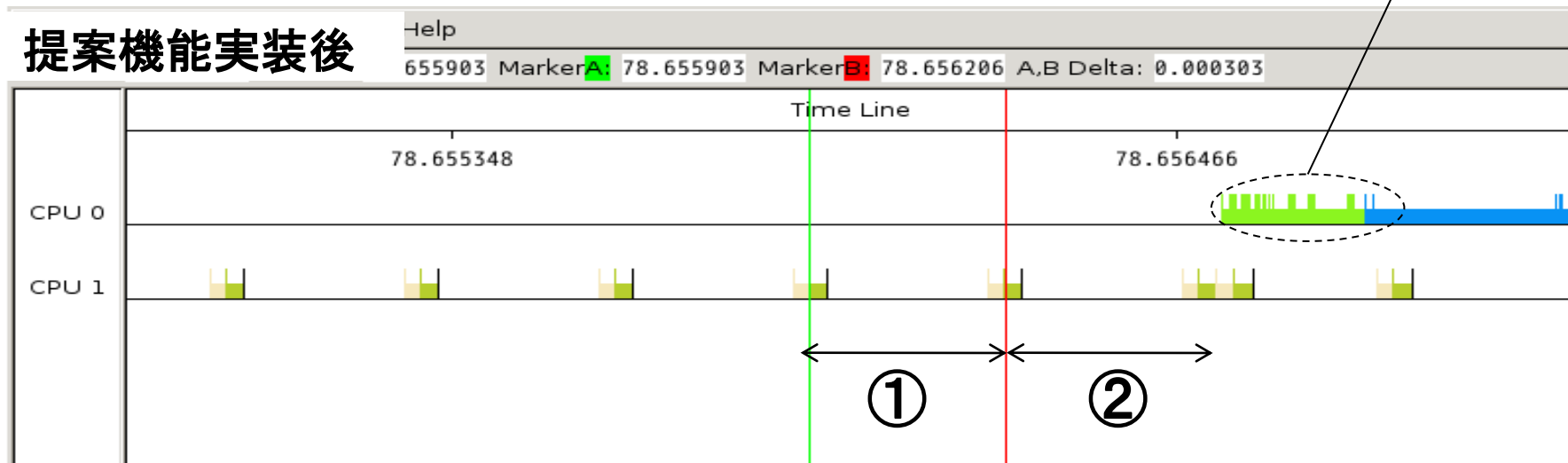
課題(ロ), (ハ)に対する提案手法



カーネルのトレースログによる評価



カスケード処理の発生



まとめ

- **単一のOS(Linux)によるリアルタイム性向上手法**
 - CPU affinity機能の適用
 - コア専用カーネルスレッドとタイマのカスケード問題
 - タイマリストを空の状態に維持する手法提案
 - タイママイグレーション
 - タイマの登録先の変更, 発火待ち
 - RT用CPUコアで, リアルタイム性阻害の要因を排除
- **今後**
 - コア専用カーネルスレッドの見直し, 更なる改善